Formularios y objetos

Pere Barnola Augé

P08/93133/01508



Índice

1. Formularios

El formulario es posiblemente la herramienta más utilizada en Internet para obtener datos e información sobre la gente que navega por la web.

La idea de los formularios es recoger información que el usuario, mediante su interacción con éstos, nos envía para, después, ejecutar una determinada acción.

Ejemplo

Los casos más comunes en Internet de formularios son, por ejemplo, los buscadores, donde el usuario introduce un texto y la web le devuelve los resultados más adecuados, o por ejemplo, cuando queremos comprar un producto y tenemos que rellenar una serie de datos personales para completar el proceso de compra.

A grandes rasgos, diremos que la información que el usuario de una web introduce en el formulario es enviada para que se procese en el servidor. El servidor es el ordenador donde la web está alojada, y será quien realice los procesos convenientes para después enviar un mensaje de respuesta al usuario. No entraremos en más detalles sobre la parte que realiza el servidor, ya que se va bastante del ámbito de esta asignatura, y no tendríamos bastante tiempo para verlo adecuadamente.

Para hacer más comprensible este punto, iremos construyendo un formulario a medida que vamos estudiando las partes, de la misma manera que hicimos en el módulo 2. Para esta tarea, crearemos un archivo nuevo con el blog de notas que guardaremos con el nombre de "formulario.html". Para empezar, introduciremos el siguiente código, con el cual crearemos un archivo XHTML básico:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>formulario contacto</title>
</head>
<body>
</body>
</html>
```




```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es"
lang="es">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<title>Formulario de contacto</title>
</head>
<body>
</body>
</html>
```

1.1. Elementos

Los elementos interactivos que tenemos a nuestra disposición para crear formularios son los siguientes: <form>, <input>, <select>, <option> y <textarea>.

Estos elementos son suficientes, como veremos, para indicar a nuestro navegador cómo tiene que recoger la información, y cómo se enviará al servidor.

1.1.1. Form

Lo primero que nuestro formulario tiene que tener es el elemento "form". Este elemento no sólo es imprescindible para la construcción de cualquier formulario, sino que todos los elementos que necesitamos para construirlo tienen que estar dentro de este elemento.

Hallamos tres posibles **atributos**:

1) Action

Este atributo indica la acción que realizará nuestro formulario. Por ejemplo, si queremos que nuestro formulario envíe un correo electrónico, el código sería una cosa así:

```
<form action=mailto:direccion@correo.com></form>
```

Si lo que queremos es que la información del formulario sea enviada al servidor para que ejecute el proceso, tenemos que indicar la URL del archivo donde se realizará este proceso.

El código lo escribiríamos de esta manera:

```
<form action="/php/proceso_formulario.php"></form>
```

2) Method

Este atributo define de qué manera los datos del formulario serán transmitidos al servidor. Las dos opciones que puede tener este atributo es "GET" o "TABLA". Más adelante, veremos en qué consiste cada una.

```
<form action="/php/proceso_formulario.php" method="POST"></form>
```

3) Enctype

Con este atributo, especificamos el tipo de encriptación de datos del formulario antes que el servidor las reciba. Sólo funcionará en el caso de que el atributo method tenga como valor "TABLA". Encriptar los datos quiere decir, a grandes rasgos, codificar estos datos antes de enviarlos al servidor.

Los tipos de valores que puede tener este atributo son:

- "**application/x-www-form-urlencoded**": es el valor por defecto si no le especificamos ninguno. Esta codificación será suficiente para la mayoría de los formularios.

- "**multipart/form-data**": especificaremos este valor en caso de que en nuestro formulario esté la posibilidad de adjuntar algún tipo de archivo.

```
<form action="/php/proceso_formulario.php" enctype="
 application/x-www-form-urlencoded"></form>
```

Siguiendo con nuestro documento de ejemplo, añadiremos el código siguiente:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>formulario contacto</title>
</head>
<body>
<form id="form1" method="post" action="proceso_formulario.php">
</form>
</body>
</html>
```

formulari.html - Bloc de notas

Archivo Edición Formato Ver Ayuda

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es"
lang="es">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<title>Formulario de contacto</title>
</head>
<body>
<form id="form1" method="post" action="proces_formulari.php">
</form>
</body>
</html>
```

1.1.2. Input

El elemento "input" es el segundo en importancia, ya que podemos decir que éste es un elemento multifunción. Esto quiere decir que con este mismo elemento se pueden generar "radiobuttons", "checkboxes" o campos para introducir texto, entre otros. Para hacerlo, al atributo "type" tendremos que especificar el tipo de elemento que queremos generar.

Los posibles valores de este atributo son: "text", "password", "checkbox", "radio", "submit", "resed"image" y "hidden". Veremos los más importantes acto seguido.

Otro atributo a tener en cuenta es "**name**". Todos los elementos de tipo "input" tienen que tener este atributo, que nos permitirá asignar un nombre al elemento.

Así nos será posible relacionar el contenido o valor de este elemento con el nombre que hemos especificado.

1.1.3. Campos de texto

Para crear una caja de texto donde el usuario pueda introducir texto, lo haremos indicando al atributo "type" el valor "text":

```
<input type="text" >
```

Los atributos que pueden tener este tipo de elementos y sus funcionalidades son los siguientes:

- **Maxlenght**. Determina el máximo de caracteres que se pueden introducir en la caja de texto.

- **Size**. Con este atributo podemos especificar la medida de la caja de texto que verá al usuario.

- **Value**. El valor que contenga este atributo indicará el texto que, por defecto, saldrá inicialmente en la caja de texto.

- **Disabled**. Nos sirve para desactivar la caja de texto. Una vez desactivada, el usuario no podrá escribir nada en el campo de texto.

- **Tabindex**. Especifica el orden del tabulador para este elemento con respecto a los otros elementos del formulario. Es decir, cada vez que el usuario haga clic en la tecla "TABULADOR" de su teclado, se activará el siguiente elemento según el orden que tenga en su atributo "tabindex".

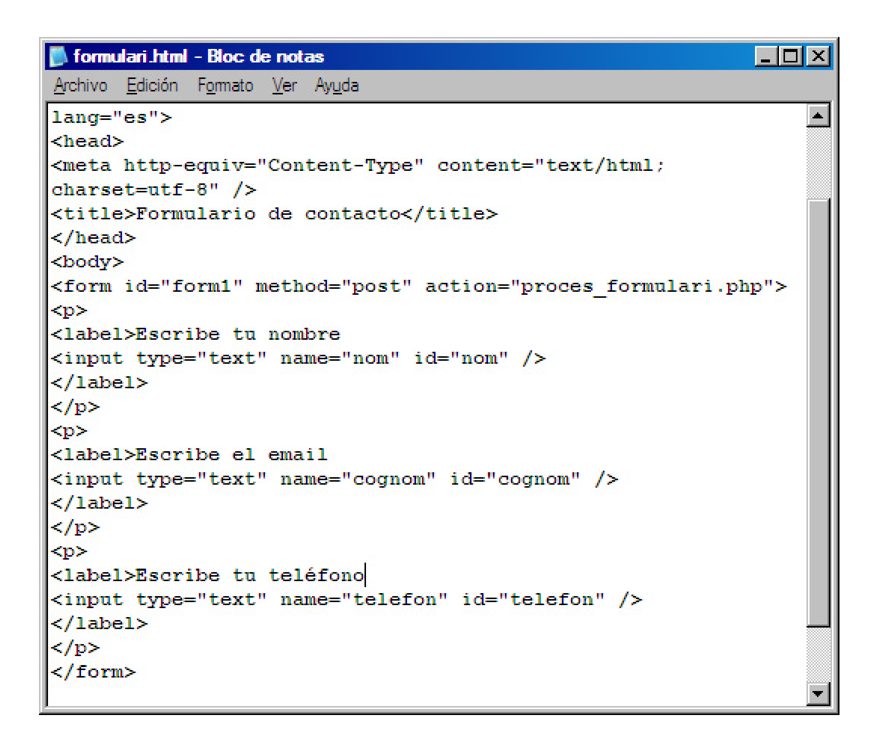
- **Alt**. Sirve para indicar una pequeña descripción al elemento.

Introduciremos unos cuantos campos de texto en nuestro documento de ejemplo:

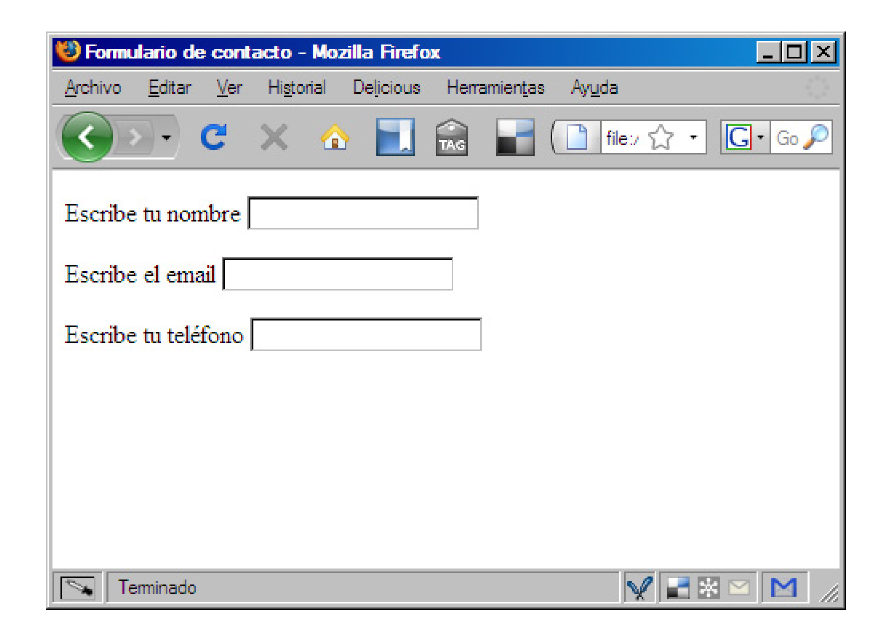
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>formulario contacto</title>
</head>
<body>
<form id="form1" method="post" action="proceso_formulario.php">
<p>
<label>Escribe tu nombre
<input type="text" name="nom" id="nom" />
</label>
</p>
<p>
<label>Escribe el email
<input type="text" name="apellido" id="apellido" />
</label>
</p>
<p>
<label>Escribe tu teléfono
<input type="text" name="telefono" id="telefono" />
</label>
</p>
</form>
</body>
```

```
</html>
```

Como vemos, en el código nos encontramos, aparte del elemento <input> que acabamos de estudiar, el elemento <label>. Este elemento lo explicaremos con detalle más adelante, pero lo incluiremos ahora para hacer más comprensible la elaboración del formulario:



Si visualizamos ahora este archivo en un navegador, lo veremos así:



Como vemos en la imagen, tenemos tres cajas de texto donde el usuario podrá introducir los diferentes datos que se piden (nombre, correo electrónico y teléfono).

1.1.4. Radiobuttons y checkboxes

Tanto los radiobuttons como los checkboxes son unos botones especiales que permiten al usuario escoger una determinada opción.

Ejemplo

Por ejemplo, si en nuestro formulario queremos hacer una pregunta al usuario del estilo "¿Tienes carné de conducir?", podremos indicarle las diferentes respuestas ("sí" y "no") con dos radiobuttons, uno para cada posible respuesta.

La **diferencia** mayor entre los radiobuttons y los checkboxes es que los primeros permiten al usuario seleccionar una entre varias opciones, mientras que los segundos dan la posibilidad de escoger una o más opciones de respuesta.

Pasaremos a comentar cada uno en detalle.

Radiobuttons

Como hemos avanzado antes, los radiobuttons permiten que el usuario sólo pueda escoger una de las posibles opciones de selección cuando el atributo "name" de los radiobuttons es el mismo. De hecho, cuando hacemos clic sobre un radiobutton, éste quedará seleccionado y el resto de radiobuttons que tengan el atributo "name" igual se deseleccionarán.

Gráficamente, los radiobuttons se visualizan como pequeños botones redondos que, una vez están seleccionados, aparecen con un pequeño icono redondo dentro.

Veremos un ejemplo de ello y lo comentaremos:

```
<p>¿Te gusta nuestra web? <br />
<label>
<input type="radio" name="gusta" id="si" value="si" checked="true" />
Si</label>
<br />
<label>
<input type="radio" name="gusta" id="no" value="no" />
No</label>
</p>
```

Como vemos, todo el conjunto de radiobuttons que forman las diferentes respuestas a una misma pregunta (en nuestro ejemplo es "¿Te gusta nuestra web?"), llevan el mismo valor para el atributo "name". No obstante, el valor del atributo "value" sí que es diferente para cada opción. Esto hará que, si seleccionamos la primera opción, el servidor recibirá la siguiente información:

[gusta=sí]

y si seleccionamos la segunda, el servidor recibirá:

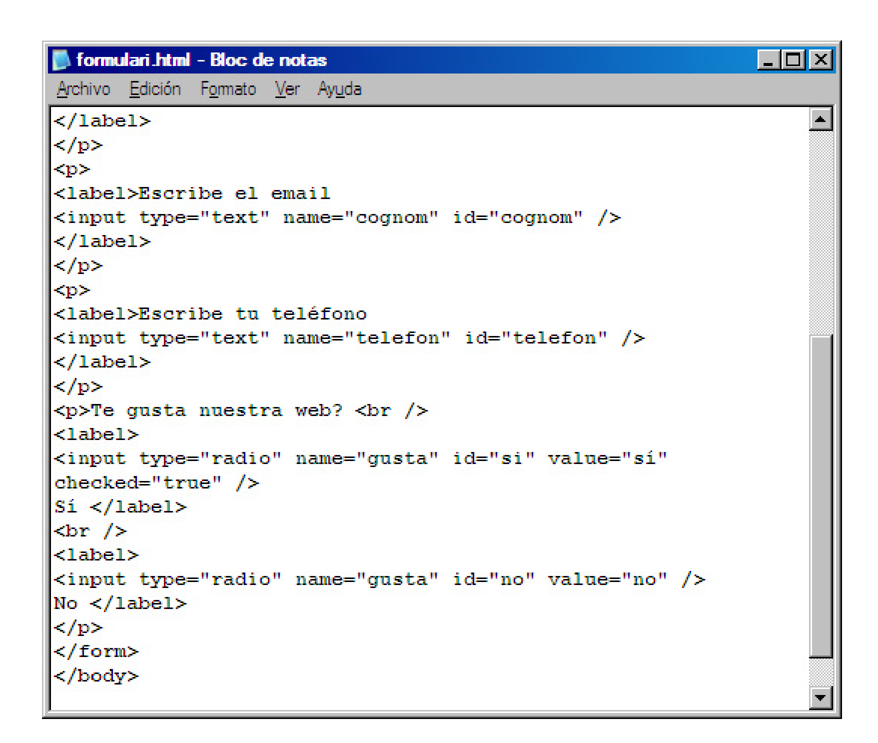
[gusta=no]

Otro atributo que vemos que presenta el primer radiobutton es "checked". Con este atributo indicamos cuál de las opciones tiene que aparecer como seleccionada por defecto, en caso de que así lo quisiéramos hacer.

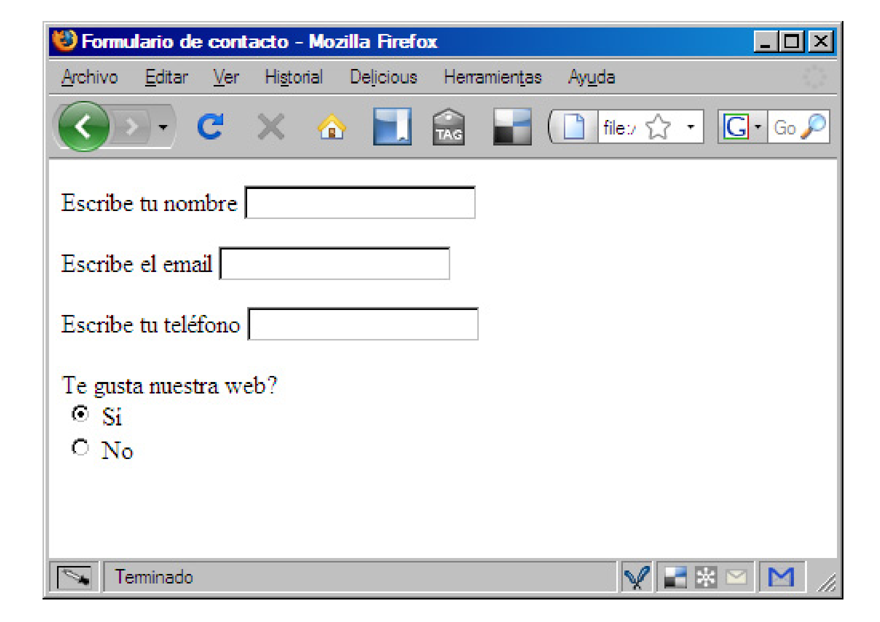
Volvamos al archivo que estamos creando y añadamos los radiobuttons:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>formulario contacto</title>
</head>
<body>
<form id="form1" method="post" action="proceso_formulario.php">
<p>
<label>Escribe tu nombre
<input type="text" name="nombre" id="nombre" />
</label>
</p>
<p>
<label>Escribe el e-mail
<input type="text" name="apellido" id="apellido" />
</label>
</p>
<p>
<label>Escribe tu teléfono
<input type="text" name="telefono" id="telefono" />
</label>
</p>
<p>¿Te gusta nuestra web? <br />
<label>
<input type="radio" name="gusta" id="si" value="si" checked="true" />
Sí</label>
```

```
<br />
<label>
<input type="radio" name="gusta" id="no" value="no" />
No</label>
</p>
</form>
</body>
</html>
```



Vemos cómo lo visualiza el navegador:



Podemos ver, tal como hemos explicado, que de las dos opciones de respuesta que damos a la pregunta "¿te gusta nuestra web?", una sale marcada por defecto con una redonda verde dentro del radiobox de la respuesta "sí".

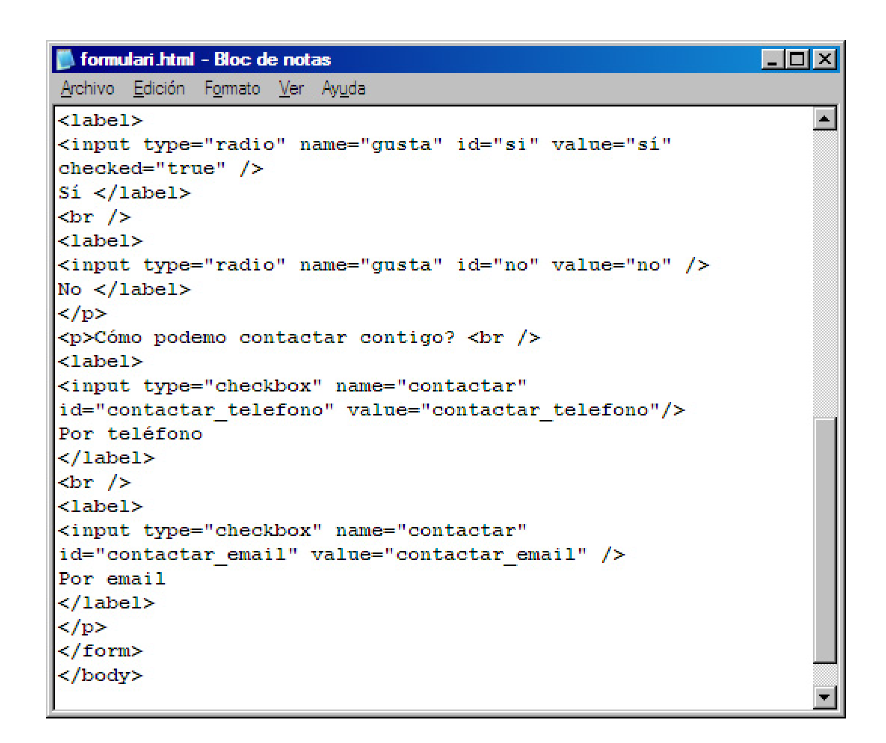
Checkboxes

Los checkboxes, a diferencia de los radiobuttons, mantienen el mismo valor para el atributo "value", pero cambian el valor del atributo "name". Es así porque las respuestas pueden ser múltiples; por lo tanto, no hay una relación entre ellas de forma excluyente.

Lo veremos claramente añadiendo unos checkboxes en nuestro archivo de ejemplo:

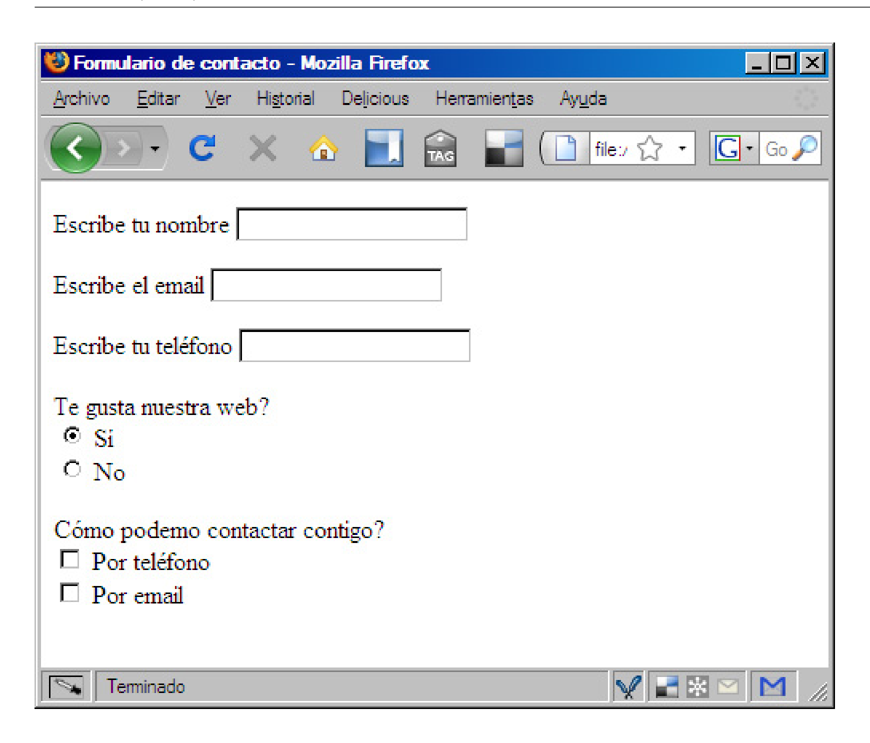
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>formulario contacto</title>
</head>
<body>
<form id="form1" method="post" action="proceso_formulario.php">
<p>
<label>Escribe tu nombre
<input type="text" name="nombre" id="nombre" />
</label>
</p>
<p>
<label>Escribe el e-mail
<input type="text" name="apellido" id="apellido" />
</label>
</p>
<p>
<label>Escribe tu teléfono
<input type="text" name="telefono" id="telefono" />
</label>
</p>
<p>¿Te gusta nuestra web? <br />
<label>
<input type="radio" name="gusta" id="si" value="si" checked="true" />
Sí</label>
<br />
<label>
<input type="radio" name="gusta" id="no" value="no" />
No</label>
```

```
</p>
<p>¿Cómo podemos contactar contigo? <br />
<label>
<input type="checkbox" name="contactar" id="contactar_telefono"
value="contactar_telefono"/>
Por teléfono
</label>
<br />
<label>
<input type="checkbox" name="contactar"
id="contactar_email" value="contactar_email" />
Por email
</label>
</p>
</form>
</body>
</html>
```

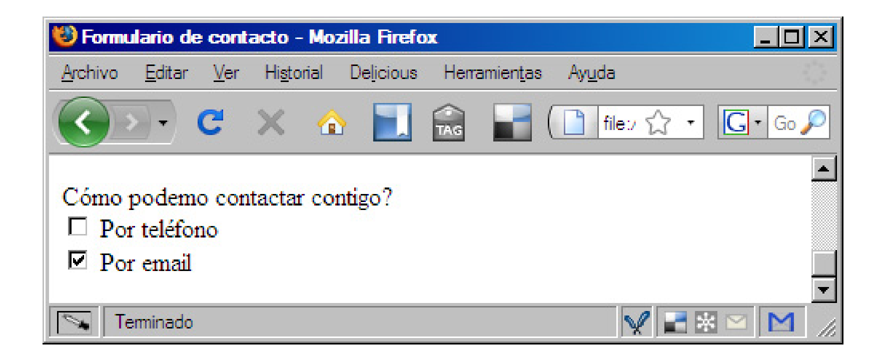



```
<label>
<input type="radio" name="gusta" id="si" value="sí"
checked="true" />
Sí </label>
<br />
<label>
<input type="radio" name="gusta" id="no" value="no" />
No </label>
</p>
<p>Cómo podemo contactar contigo? <br />
<label>
<input type="checkbox" name="contactar"
id="contactar_telefono" value="contactar_telefono"/>
Por teléfono
</label>
<br />
<label>
<input type="checkbox" name="contactar"
id="contactar_email" value="contactar_email" />
Por email
</label>
</p>
</form>
</body>
```

Y si lo visualizamos en un navegador tendríamos este resultado:



Como veremos en la imagen, la diferencia entre los checkboxes y los radiobuttons también se ve en lo visual. Los checkboxes aparecen como botones cuadrados que, cuando están seleccionados, se muestran con una "marca" dentro. Lo vemos en la siguiente imagen:



De la misma manera que hemos visto con los radiobuttons, podemos hacer uso del atributo "checked" para marcar una opción por defecto.

1.1.5. Botones de acción

Los botones de acción son útiles en los formularios para enviar los datos, ejecutar funciones personalizadas, etc.

Los botones de acción más comunes en los formularios son los de **submit** (envío de los datos) y los de **reset** (restablece los campos del formulario a sus valores por defecto). Veremos en profundidad estos dos tipos.

Submit

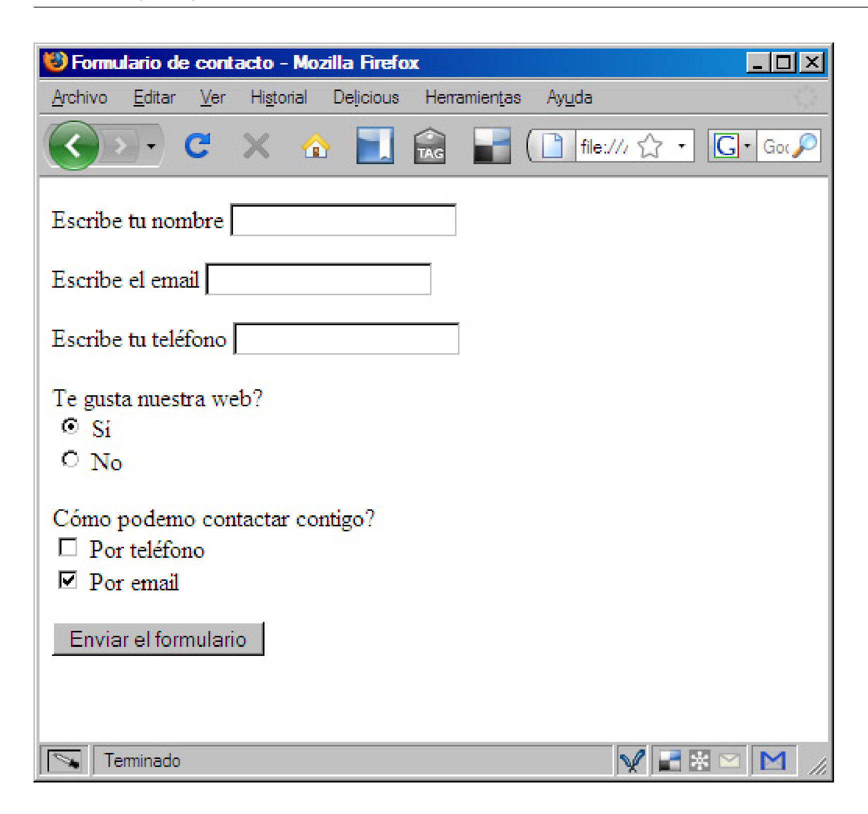
Este es el botón que permitirá enviar automáticamente la información del formulario cuando hagamos clic encima. Para conseguir un botón de este tipo tenemos que dar al atributo "type" el valor de "submit", tal como vemos en el ejemplo.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>formulario contacto</title>
</head>
<body>
<form id="form1" method="post" action="proceso_formulario.php">
<p>
<label>Escribe tu nombre
<input type="text" name="nombre" id="nombre" />
</label>
</p>
<p>
<label>Escribe el e-mail
<input type="text" name="apellido" id="apellido" />
</label>
</p>
<p>
<label>Escribe tu teléfono
<input type="text" name="telefono" id="telefono" />
</label>
</p>
<p>¿Te gusta nuestra web? <br />
<label>
<input type="radio" name="gusta" id="si" value="si" checked="true" /> Sí</label>
<br />
<label>
<input type="radio" name="gusta" id="no" value="no" />
No</label>
</p>
<p>¿Cómo podemos contactar contigo? <br />
<label>
<input type="checkbox" name="contactar" id="contactar_telefono" value="contactar_telefon"/>
Por teléfono
</label>
<br />
<label>
```

```
<input type="checkbox" name="contactar" id="contactar_email" value="contactar_email" />
Por email
</label>
</p>
<p>
<input type="submit" name="submit" id="submit" value="Enviar el formulario" />
</p>
</form>
</body>
</html>
```

```
formulari.html - Bloc de notas
Archivo Edición Formato Ver Ayuda
<br />
<label>
<input type="radio" name="gusta" id="no" value="no" />
No </label>
</p>
<p>Cómo podemo contactar contigo? <br />
<label>
<input type="checkbox" name="contactar"
id="contactar_telefono" value="contactar_telefono"/>
Por teléfono
</label>
<br />
<label>
<input type="checkbox" name="contactar"
id="contactar_email" value="contactar_email" />
Por email
</label>
</p>
<p>
<input type="submit" name="submit" id="submit" value="Enviar
el formulario" />
</p>
</form>
</body>
```

Vemos cómo interpreta nuestro código el navegador:



Con el atributo "value", especificaremos el texto que queremos que aparezca dentro del botón de "submit", en nuestro caso "Enviar el formulario".

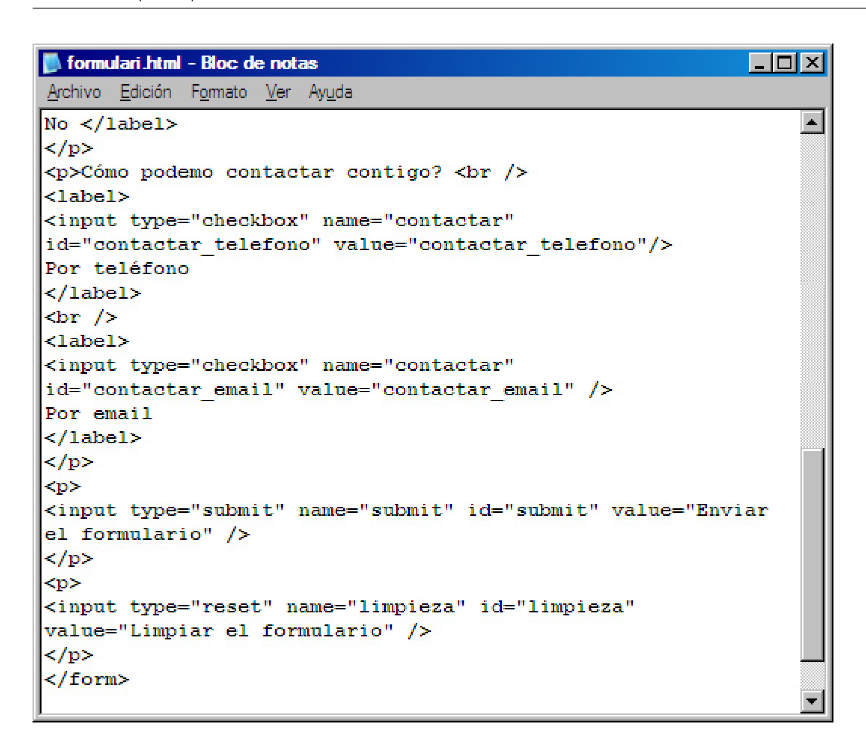
Reset

Es posible que el usuario, una vez haya rellenado el formulario, quiera reiniciarlo y volver a escribir la información. Para esta tarea utilizaremos el botón de "reset", que sirve para restablecer los campos del formulario a sus valores por defecto.

Para crear un botón "reset", lo haremos de la forma siguiente:

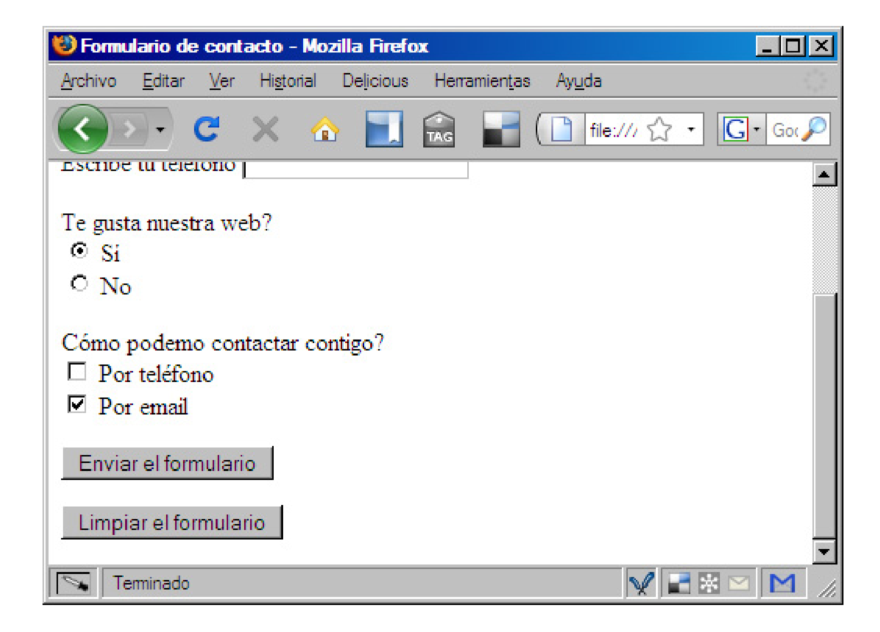
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>formulario contacto</title>
</head>
<body>
<form id="form1" method="post" action="proceso_formulario.php">
<p>
<label>Escribe tu nombre
<input type="text" name="nom" id="nom" />
</label>
</p>
<p>
```

```
<label>Escribe el e-mail
<input type="text" name="apellido" id="apellido" />
</label>
</p>
<p>
<label>Escribe tu teléfono
<input type="text" name="telefono" id="telefono" />
</label>
</p>
<p>¿Te gusta nuestra web? <br />
<label>
<input type="radio" name="gusta" id="si" value="si" checked="true" />
Sí</label>
<br />
<label>
<input type="radio" name="gusta" id="no" value="no" />
No</label>
</p>
<p>¿Cómo podemos contactar contigo? <br />
<label>
<input type="checkbox" name="contactar" id="contactar_telefono" value="contactar_telefono"/>
Por teléfono
</label>
<br />
<label>
<input type="checkbox" name="contactar" id="contactar_email" value="contactar_email" />
Por email
</label>
</p>
<p>
<input type="submit" name="submit" id="submit" value="Enviar el formulario" />
</p>
<p>
<input type="reset" name="limpieza" id="limpieza" value="Limpiar el formulario" />
</p>
</form>
</body>
</html>
```



```
No </label>
</p>
<p>Cómo podemo contactar contigo? <br />
<label>
<input type="checkbox" name="contactar"
id="contactar_telefono" value="contactar_telefono"/>
Por teléfono
</label>
<br />
<label>
<input type="checkbox" name="contactar"
id="contactar_email" value="contactar_email" />
Por email
</label>
</p>
<p>
<input type="submit" name="submit" id="submit" value="Enviar
el formulario" />
</p>
<p>
<input type="reset" name="limpieza" id="limpieza"
value="Limpiar el formulario" />
</p>
</form>
```

Vemos el resultado de nuestro código en el navegador:



1.1.6. Listas de opciones o desplegables

Las listas de opciones son otra manera de dar al usuario la opción de escoger una o varias respuestas.

Veremos un ejemplo de ello y lo comentaremos:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
```

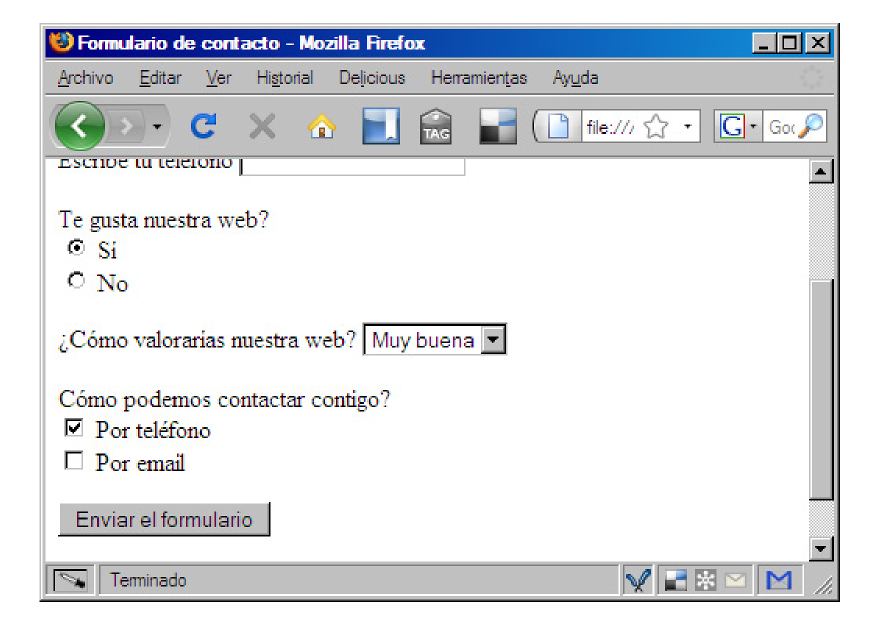
```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>formulario contacto</title>
</head>
<body>
<form id="form1" method="post" action="proceso_formulario.php">
<p>
<label>Escribe tu nombre
<input type="text" name="nombre" id="nombre" />
</label>
</p>
<p>
<label>Escribe el e-mail
<input type="text" name="apellido" id="apellido" />
</label>
</p>
<p>
<label>Escribe tu teléfono
<input type="text" name="telefono" id="telefono" />
</label>
</p>
<p>¿Te gusta nuestra web? <br />
<label>
<input type="radio" name="gusta" id="si" value="si" checked="true" />
Sí</label>
<br />
<label>
<input type="radio" name="gusta" id="no" value="no" />
No</label>
</p>
<p>
¿<label>Cómo valorarías nuestra web?
<Select name="valoracion" id="valoracion">
<option value="10" selected="selected">muy buena</option>
<option value="7">buena</option>
<option value="5" >normal</option>
<option value="3">mala</option>
<option value="1">muy mala</option>
</Select>
</label>
</p>
<p>¿Cómo podemos contactar contigo? <br />
<label>
<input type="checkbox" name="contactar" id="contactar_telefono" value="contactar_telefono"/>
Por teléfono
</label>
<br />
```

```
<label>
<input type="checkbox" name="contactar" id="contactar_email" value="contactar_email" />
Por email
</label>
</p>
<p>
<input type="submit" name="submit" id="submit" value="Enviar el formulario" />
</p>
<p>
<input type="submit" name="limpieza" id="limpieza" value="Limpiar el formulario" />
</p>
</form>
</body>
</html>
```

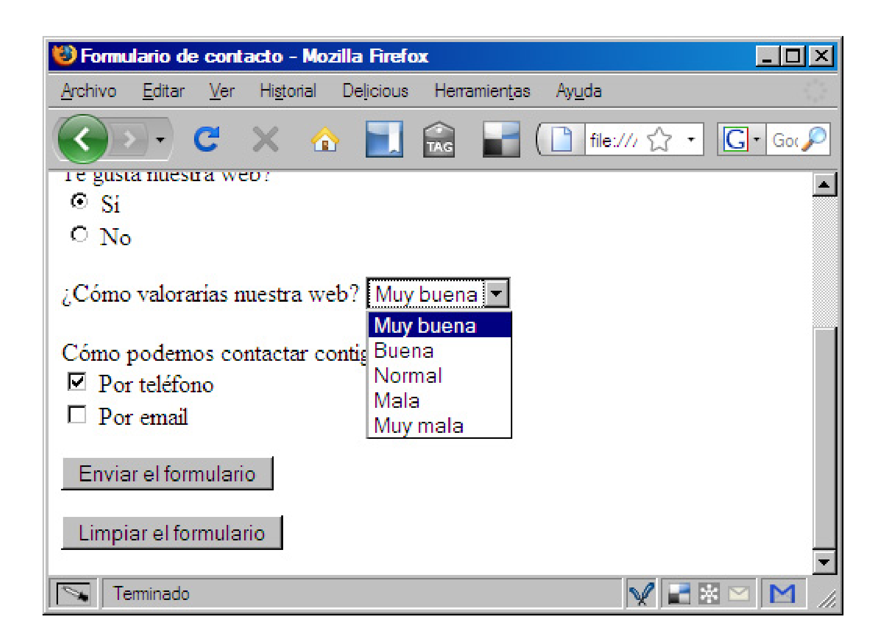
El par de etiquetas `<select>` definen nuestra lista. Una vez creada, iremos introduciendo las diferentes opciones que el usuario tiene que escoger. Eso lo haremos mediante el elemento `<option>`. Vemos que cada elemento `<option>` tiene un atributo "value" diferente, mientras que el elemento `<select>` tiene un atributo "name". Eso quiere decir que, cuando enviamos la información del formulario, el atributo "name" se asociará con el valor del atributo "value" del elemento que el usuario haya escogido. Es decir, si el usuario escoge la segunda opción, el formulario se enviará con la siguiente información:

[valoracion=7]

Vemos en la imagen siguiente cómo se visualizaría nuestra lista:

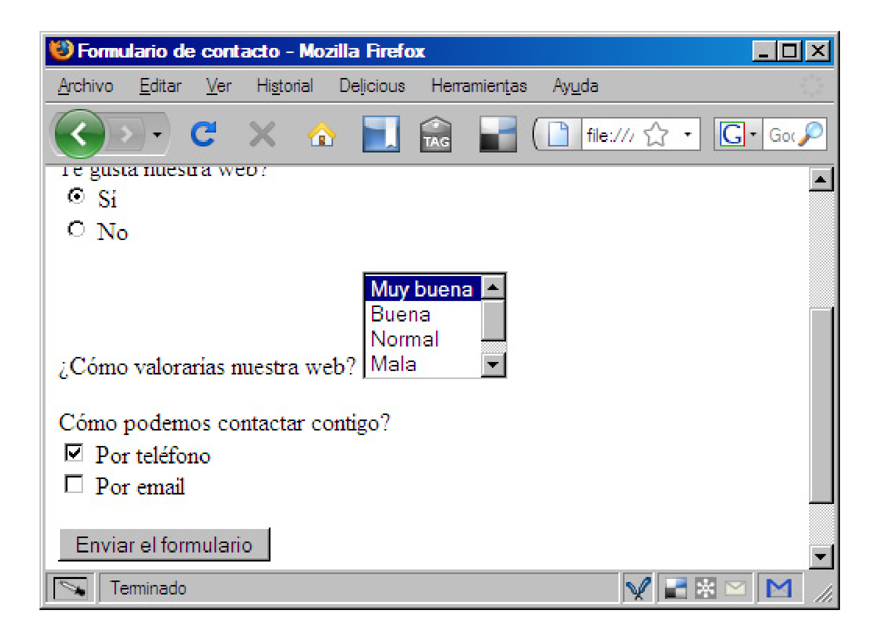


Como podemos ver en la imagen, las listas, por defecto, vienen juntas y una vez hacemos clic encima, se despliegan y muestran todas las opciones que tienen. Veamos en la imagen siguiente el efecto que tiene sobre nuestra lista cuándo hacemos clic encima:



Si queremos que nuestras listas aparezcan siempre desplegadas, añadiremos el atributo "size" al elemento "select". El valor que indicamos en este atributo delimitará la altura que tendrá nuestra lista. Veámoslo en un ejemplo:

```
¿<label>Cómo valorarías nuestra web?
<Select name="valoracion" id="valoracion" size=4>
<option value="10" selected="selected">muy buena</option>
<option value="7">buena</option>
<option value="5" >normal</option>
<option value="3">mala</option>
<option value="1">muy mala</option>
</Select>
</label>
```



Como vemos en la imagen, la altura de nuestra lista equivale a cuatro elementos. Si hay más opciones de las especificadas en el atributo "size", como es el caso de nuestro ejemplo, saldrá un pequeño scroll en la parte derecha que nos permitirá acceder a todas las opciones.

1.1.7. Etiquetas (label)

Las etiquetas son los textos que describen cada elemento que forma parte de un formulario. Por lo tanto, siempre irán asociadas a otros elementos. Estos elementos recibirán el nombre de controles.

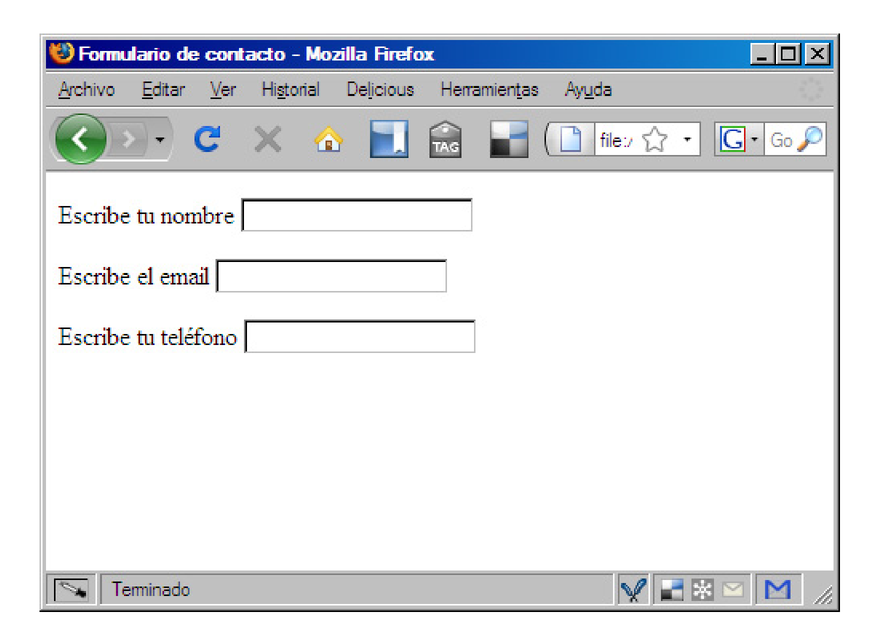
Las etiquetas sirven para ayudar al usuario a entender la función de cada control. Entre otras cosas, describen el contenido de los campos de texto, y dan nombre a los valores de los checkboxes y radiobuttons.

Al principio de este capítulo hemos comentado que utilizaríamos etiquetas para hacer más entendedor nuestro ejemplo. Recordemos el código que vamos a utilizar:

```
<p>
<label>Escribe tu nombre
<input type="text" name="nombre" id="nombre" />
</label>
</p>
<p>
<label>Escribe el e-mail
<input type="text" name="apellido" id="apellido" />
</label>
</p>
<p>
<label>Escribe tu teléfono
```

```
<input type="text" name="telefono" id="telefono" />
</label>
</p>
```

El resultado que entonces obteníamos en el navegador era así:



1.1.8. Agrupación de campos (fieldset)

Este elemento nos permite agrupar visualmente una serie de elementos del formulario. Es decir, si en nuestro formulario tuviéramos dos partes que quisiéramos diferenciar de manera visual las podríamos englobar con dos elementos "fieldset".

Por defecto, el impacto visual que genera un fieldset es la aparición de un rectángulo que engloba los elementos dentro del fieldset. Este rectángulo puede llevar un título para hacer más comprensible la diferenciación. Este título lo especificaremos con el elemento "legend", que puede admitir un atributo "align" que nos permitirá indicar la posición del título. Los valores que puede soportar "align" son "left", "center", "right, top" y "bottom".

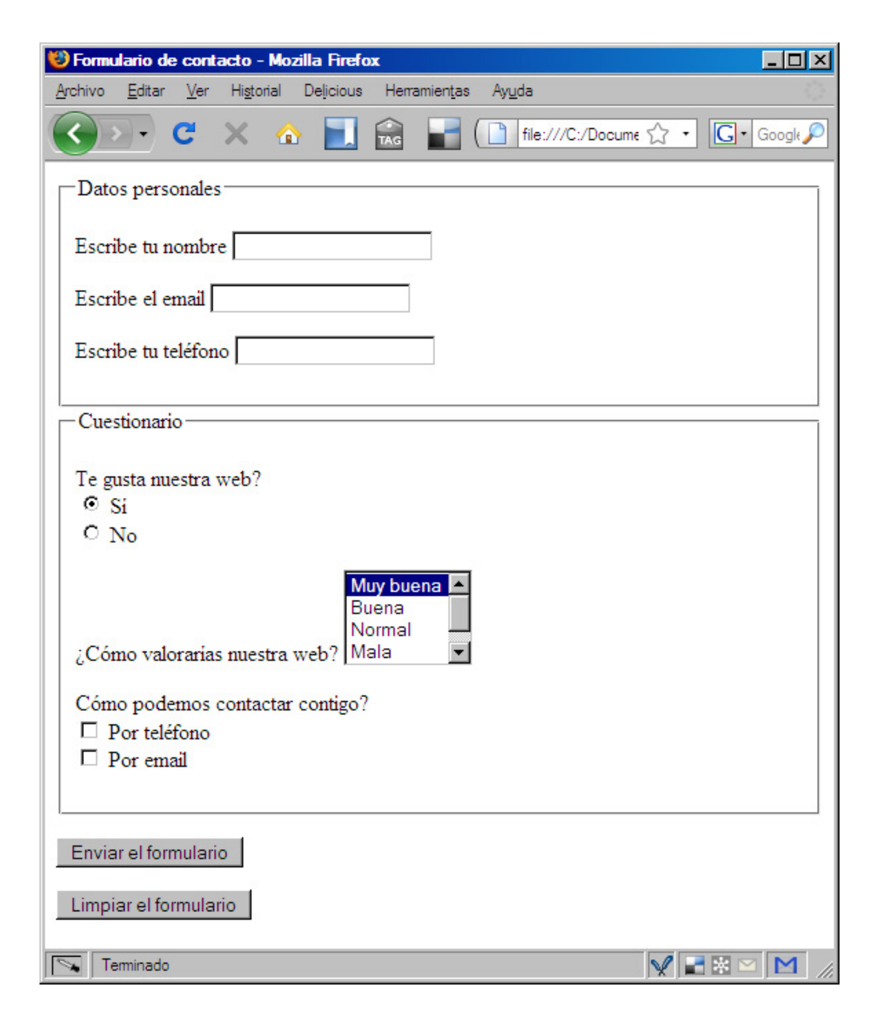
Continuando con nuestro ejemplo, agruparemos el formulario que hemos creado con dos partes. Una para la parte donde pedimos los datos y otra para la parte del pequeño cuestionario:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>formulario contacto</title>
```

```
</head>
<body>
<form id="form1" method="post" action="proceso_formulario.php">
<fieldset>
<legend>Datos personales</legend>
<p>
<label>Escribe tu nombre
<input type="text" name="nombre" id="nombre" />
</label>
</p>
<p>
<label>Escribe el e-mail
<input type="text" name="apellido" id="apellido" />
</label>
</p>
<p>
<label>Escribe tu teléfono
<input type="text" name="telefono" id="telefono" />
</label>
</p>
</fieldset>
<fieldset>
<legend>Cuestionario</legend>
<p>¿Te gusta nuestra web? <br />
<label>
<input type="radio" name="gusta" id="si" value="si" checked="true" />
Sí</label>
<br />
<label>
<input type="radio" name="gusta" id="no" value="no" />
No</label>
</p>
<p>
¿<label>Cómo valorarías nuestra web?
<Select name="valoracion" id="valoracion">
<option value="10" selected="selected">muy buena</option>
<option value="7">buena</option>
<option value="5" >normal</option>
<option value="3">mala</option>
<option value="1">muy mala</option>
</Select>
</label>
</p>
<p>¿Cómo podemos contactar contigo? <br />
<label>
<input type="checkbox" name="contactar" id="contactar_telefono" value="contactar_telefono"/>
Por teléfono
```

```
</label>
<br />
<label>
<input type="checkbox" name="contactar" id="contactar_email" value="contactar_email" />
Por email
</label>
</p>
</fieldset>
<p>
<input type="submit" name="submit" id="submit" value="Enviar el formulario" />
</p>
<p>
<input type="submit" name="limpieza" id="limpieza" value="Limpiar el formulario" />
</p>
</form>
</body>
</html>
```

Vemos el resultado de este código:



1.2. Los métodos TABLA y GET

Al principio del capítulo, hemos dicho que los datos del formulario se envían al servidor mediante el método indicado al atributo "method" del elemento "form". Pues bien, los dos métodos posibles son "GET" y "POST".

La diferencia entre estos dos métodos radica en la forma de enviar los datos del formulario.

El **método** "**GET**" envía los datos mediante la URL.

Ejemplo

Por ejemplo, si enviamos un formulario con este método veríamos en la barra de dirección una URL de este estilo:

http://url?parametre1=valor1&parametre2=valor2

Este método tiene como inconveniente la limitación de la URL. Es decir, si tenemos un formulario con bastantes campos y en el que el usuario puede introducir grandes cantidades de texto, es muy probable que sobrepasemos el límite de 256 caracteres que, como máximo, puede soportar la URL.

Una ventaja de este método es que podemos construir enlaces con los valores ya introducidos. Este sistema resulta bastante útil cuando necesitamos comprobar el funcionamiento del formulario y queremos evitar que se tenga que llenar continuamente.

Con el **método** "**POST**" los datos son enviados de manera invisible por el usuario, ya que la información del formulario se almacena en una parte del servidor. Además, si utilizamos el método "TABLA" no tendremos ninguna restricción con respecto a la cantidad de información que podemos enviar.

Actividad. Un formulario de contacto

La práctica de este apartado consiste en construir un formulario que contenga todos los elementos que hemos ido viendo.

2. Objetos

Actualmente, además de texto e imágenes, en la web se encuentran a menudo vídeos, sonidos o elementos interactivos. Estos contenidos, independientemente del formato, se identifican como objetos. La etiqueta que los representa es, por lo tanto, "object".

2.1. Propiedades y definición

Los objetos dependerán siempre de un conector (plug-in) para ser reproducidos correctamente en el navegador.

Ejemplo

Un ejemplo generalizado del uso de conectores es el reproductor de archivos .swf, más conocido por "Flash Player" o reproductor de archivos flash. El hecho de no tenerlo instalado en el navegador hará imposible la visualización de archivos flash.

El W3C recomienda el uso del elemento object para incluir estos objetos, declarando, dentro del elemento object, de qué tipo se trata por medio del atributo type, que indicará al navegador cuál es el tipo de conector que tiene que utilizar, su ubicación con el atributo data, así como los parámetros del objeto con los valores respectivos.

2.2. Ejemplos comunes

Los ejemplos más comunes a la hora de incluir objetos son para visualizar películas flash interactivas.

Para incorporar una de estas películas en nuestra web, tendríamos que especificar el código siguiente:

```
<object type="application/x-shockwave-flash" data="prova.swf" width="550" height="450">
<param name="quality" value="high" />
<p>No tienes el conector de Flash instalado en el navegador</p>
</object>
```

Como vemos, tendremos que tener un archivo "prova.swf" en nuestro disco duro, con el fin de poder visualizarlo. Si no tenemos el plugin de Flash instalado en nuestro navegador, se visualizará la parte dentro del elemento object; en nuestro caso, se visualizaría el párrafo siguiente:

```
<p>No tienes el conector de Flash instalado en el navegador</p>
```